

Ruby - Bug #14046

Ripper loses DATA

10/24/2017 08:15 AM - mjago (Martyn Jago)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Target version:</b>		
<b>ruby -v:</b>	ruby 2.5.0dev (2017-10-19 trunk 60213) [x86_64-darwin13]	<b>Backport:</b> 2.3: UNKNOWN, 2.4: UNKNOWN

Description

Since the following is a valid self-contained script:

```
p DATA.read # comment
__END__
abc
```

should the \_\_END\_\_ suffix data be preserved with :on\_\_end\_\_ ?  
Alternatively should there be an :on\_ignored\_\_end token to preserve DATA?  
Notably the comment content is preserved:

```
[[[1, 0], :on_ident, "p", EXPR_CMDARG],
 [[1, 1], :on_sp, " ", EXPR_CMDARG],
 [[1, 2], :on_const, "DATA", EXPR_ARG],
 [[1, 6], :on_period, ".", EXPR_DOT],
 [[1, 7], :on_ident, "read", EXPR_ARG],
 [[1, 11], :on_sp, " ", EXPR_ARG],
 [[1, 12], :on_comment, "# comment\n", EXPR_ARG],
 [[2, 0], :on__end__, "__END__\n", EXPR_BEG]]
```

History

#1 - 03/06/2021 12:39 AM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

Ruby stops parsing the file when \_\_END\_\_ is reached. Since the parser doesn't see the rest of the file, it is expected that Ripper doesn't either. If you want to get DATA content, you can handle that using the result of the lex:

```
require 'ripper'

s = "p DATA.read # comment\n__END__\n abc\n"

lex = Ripper.lex(s)

if lex[-1][1] == :on__end__
  off = -1
  lex[-1][0][0].times{off = s.index("\n", off+1)}
  data = s[(off+1)..-1]
end
```